Techniken der Projektentwicklung

Swing Concepts

Ingo Lütkebohle

Termin 11

Einordnung

Was bisher war

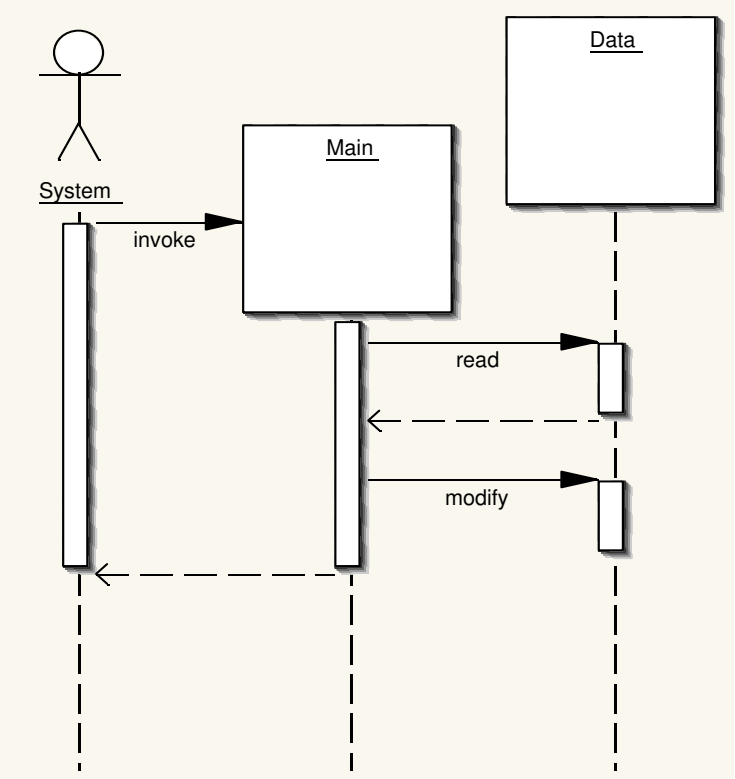
- Allgemeine Entwurfstechniken
- Serielle Programmabläufe (AuD)

Thema heute

- Architektur grafischer Benutzeroberflächen (GUIs)
- GUIs in Java mit Swing

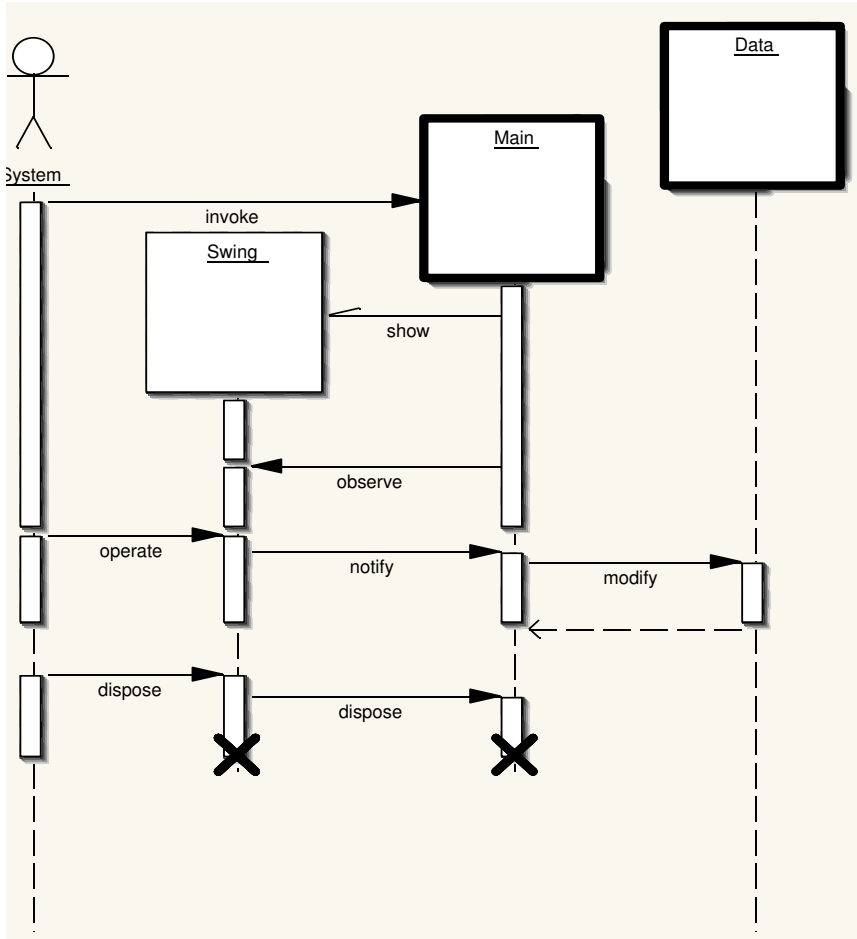
Serielle Programmarchitektur

- Programm kontrolliert Ablauf
- Eingabe-Verarbeitung-Ausgabe
- Steuerung: Start, festgelegte Punkte
- Typisch für Übungsaufgaben, kleine Skripte, etc.



Architektur grafischer Programme

- System (Benutzer) kontrolliert Ablauf
- Ablauf variabel
- Steuerung **ereignisorientiert**, frei
- *Inversion of Control*



Beispielhafte Ereignisse

Neuzeichnen bei...

- Aufdecken des Fensters
- Selektion eines Elements

Eingabe annehmen...

- Drücken eines Buttons
- Texteingabe

Benachrichtigungsmechanismen

Direkte Methodenaufrufe

- z.B. `paintComponent`
- nur für sichtbare Komponenten
- geschieht automatisch

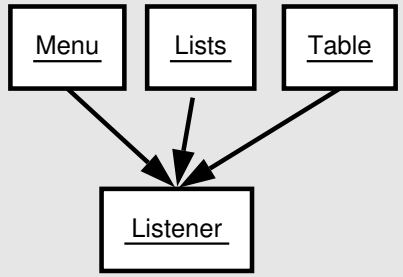
Ereignisbenachrichtigung

1. Entwickler implementiert `EventListener`
2. Listener-Instanz an GUI-Komponente anmelden
3. Komponente ändert Zustand
4. Listener-Methode empfängt `Event`

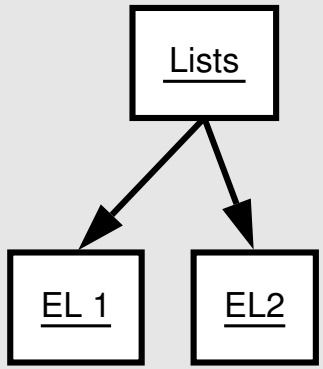
Vorteile von Event-Listenern

- Komponente selbst immer gleich
- Gruppierung logisch zusammenhängender Ereignisse mehrerer Komponenten
- Aufteilung unzusammenhängender Bearbeitung von Ereignissen einer Komponente

Gruppierung



Aufteilung

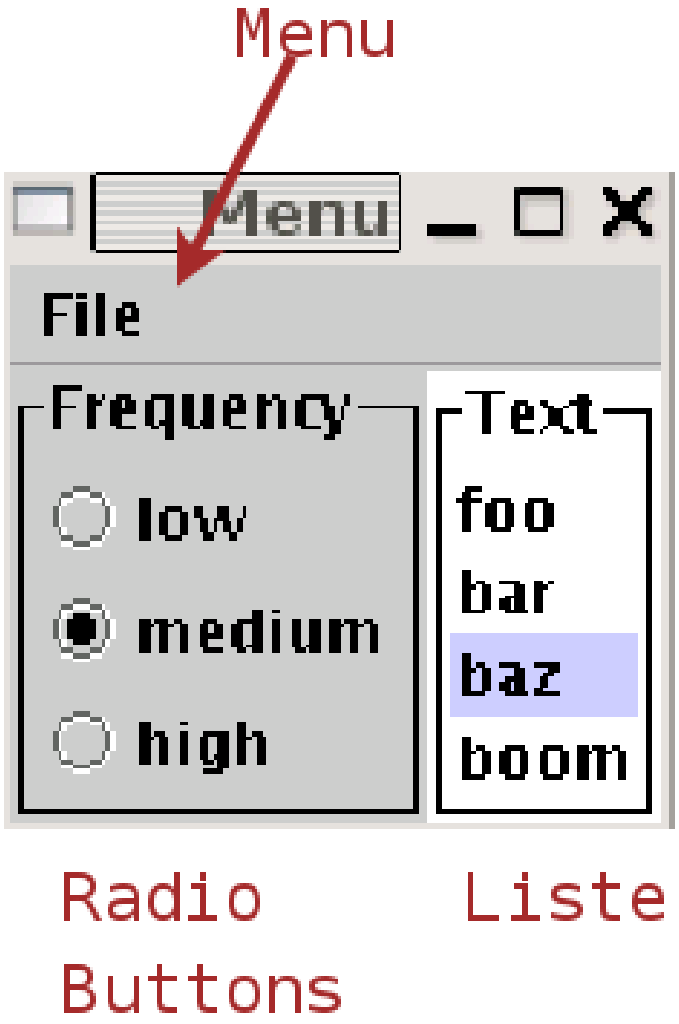


Hinweise zur Implementierung von EventListenern

- Initial: Ein EventListener pro Use Case
- Reiner *Vermittler* zwischen GUI und Rest
- Für Swing zeitkritisch (da im Swing-Thread aufgerufen)
- Falls Neuzeichnen der GUI notwendig, `repaint` aufrufen.

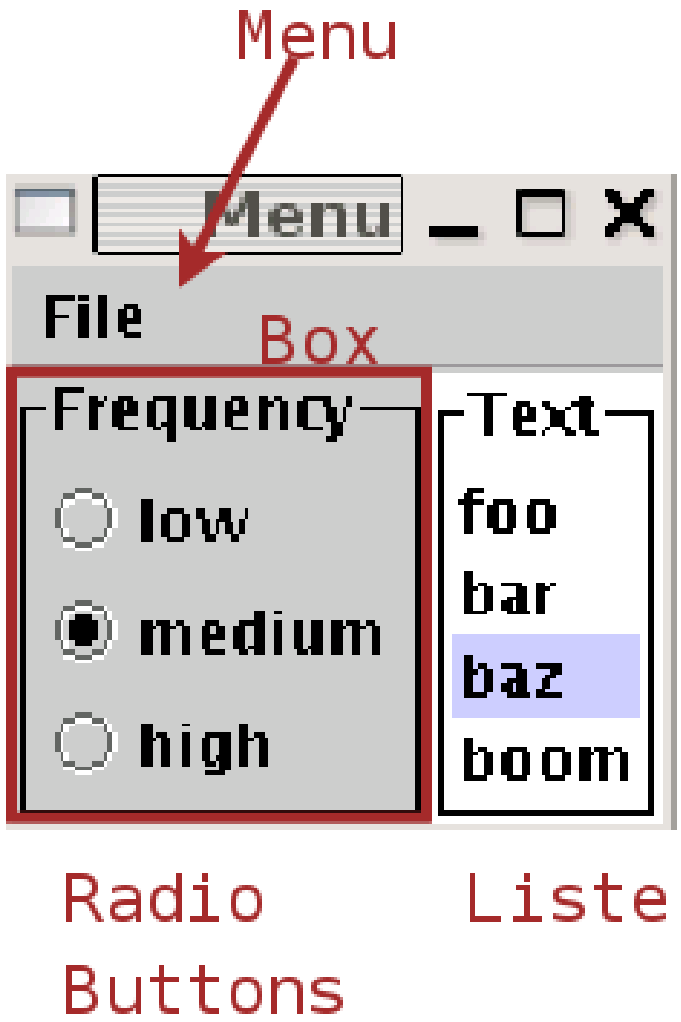
Swing Baukasten

- Komponenten für Benutzerinteraktion
  → Direkt verwendbar
- Container zur Organisation
- Layout-Manager zur Anordnung
  (hier: vertical/horizontal Box)



Swing Baukasten

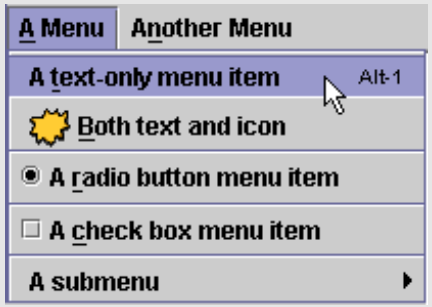
- Komponenten für Benutzerinteraktion
  → Direkt verwendbar
- Container zur Organisation
- Layout-Manager zur Anordnung
  (hier: vertical/horizontal Box)



Wichtige Komponenten: Schnittstellenelemente

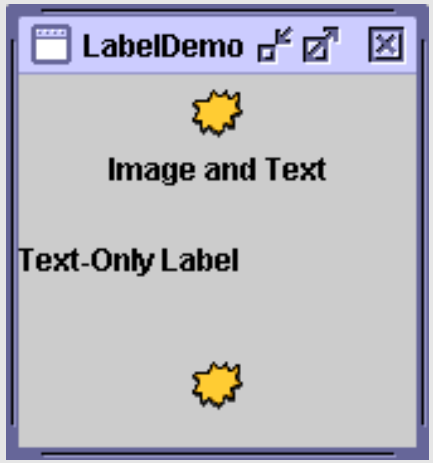
JTextfield

JMenu & -Item

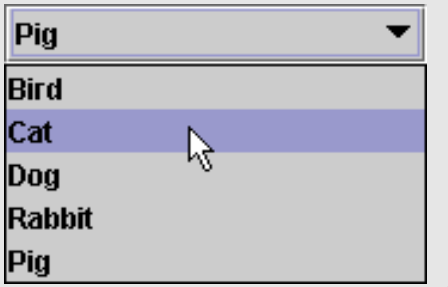


JButton

JLabel



JComboBox



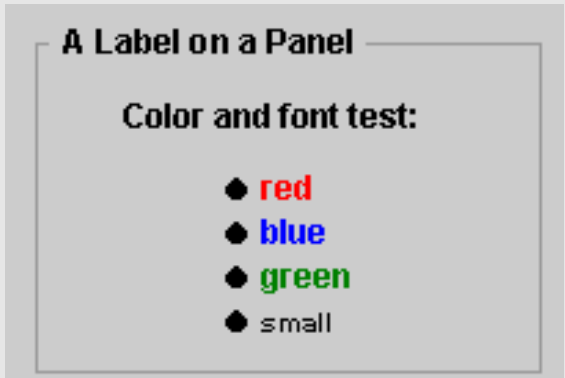
JSlider

Bilder aus http://java.sun.com/docs/books/tutorial/uiswing/components/components.html

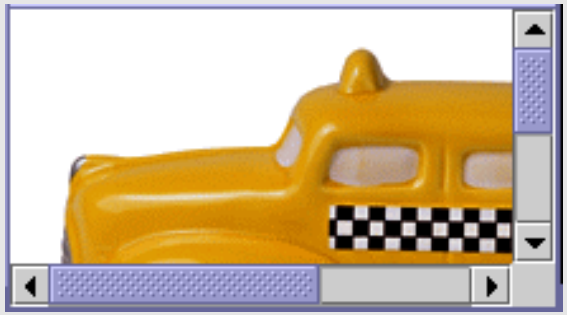
Wichtige Komponenten: Container

JFrame

JPanel

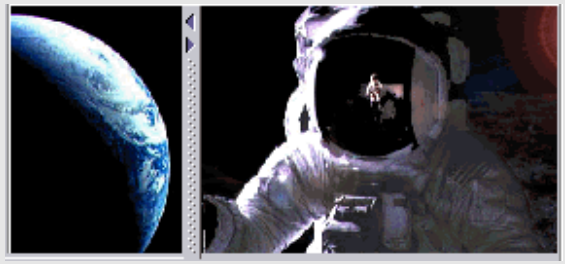


JScrollPane



JTabbedPane

JSplitPane



Bilder aus http://java.sun.com/docs/books/tutorial/uiswing/components/components.html

Warum Layout-Manager?

Früher: Manuelle, pixelgenaue Anordnung

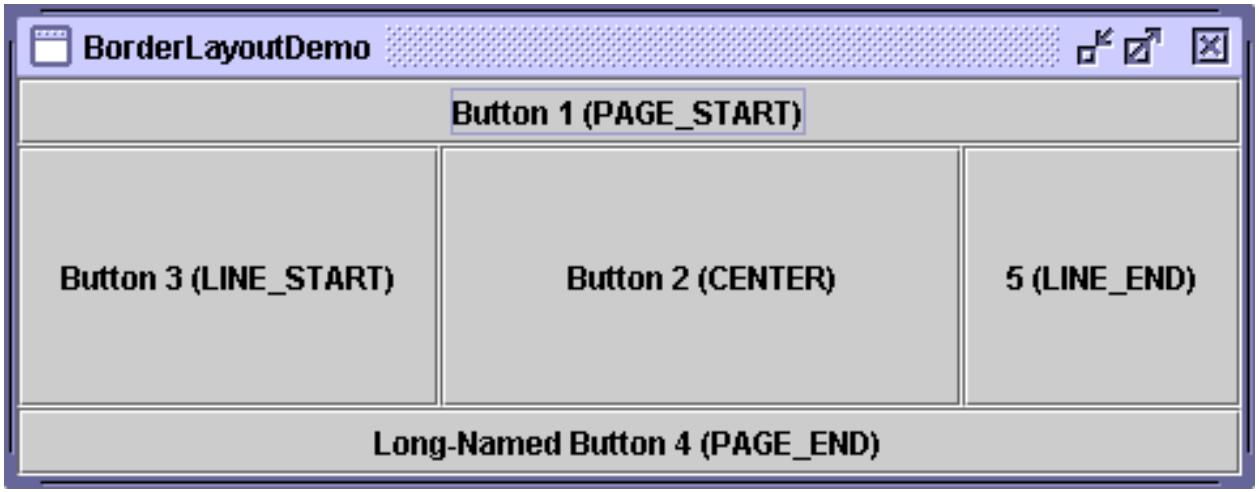
Probleme:

- Textabmessungen variieren (Font, Sprache)
- Dynamische Erzeugung von GUIs
- Hoher Änderungsaufwand

In Swing: Relatives Layout

1. Relative grobe Positionsangaben („zuoberst, rechts von“)
2. Komponenten kennen ihre Größe
3. Layout-Manager berechnet den Rest

BorderLayout

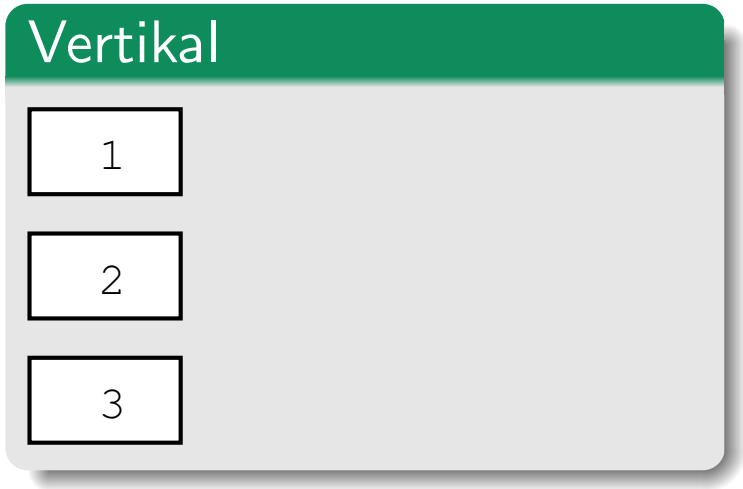


- Default für `JFrame ContentPane`
- Feste Positionen für jeweils *genau eine* Komponente
- Gibt `CENTER` allen Platz der nicht sonst benötigt wird
- Nicht alle Positionen müssen besetzt werden

http://java.sun.com/docs/books/tutorial/uiswing/layout/border.html

BoxLayout und Box

- Anordnungen:
  - Vertikal: links → rechts
  - Horizontal: oben → unten
- Beachtet Komponentengrößen
- Sehr weit anpassbar
- Box-Klasse mit praktischen Hilfsfunktionen



http://java.sun.com/docs/books/tutorial/uiswing/layout/box.html

Eigene Komponenten

- Ableiten von JComponent oder Unterklasse
  $\longrightarrow$ JPanel ist ganz praktisch
- Muss Zeichenfläche selbst verwalten
- Kommunikation mit externen Komponenten sollte Listener-Konzept folgen
- Kann eigene Events/Listener definieren

Zeichnen auf Anforderung

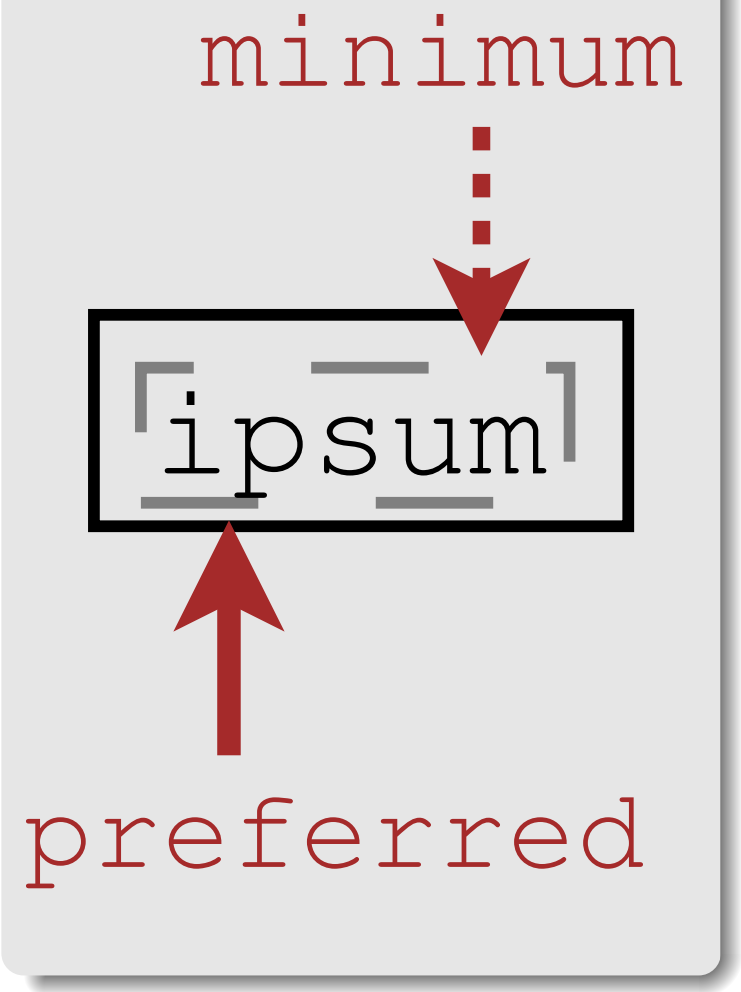
- `paintComponent` überladen
- Muss jederzeit alles zeichnen können
- Kann ggfls. kleinere Bereiche zeichnen ("clipping rectangle")
- Zeitkritisch (da im UI-Thread)
  $\longrightarrow$ keine großen Berechnungen!
- Standardmäßig gepufferte Anzeige (kein "flicker")

http://java.sun.com/docs/books/tutorial/uiswing/14painting/concepts.html

Interaktion mit dem Layout-Manager

- Größe über `setPreferredSize` vorgeben
- In `paintComponent` tatsächliche Größe abfragen
- Dokumentation des Layout-Managers für Spezifika beachten

Beispielgrößen



Hello World

- Erzeugen eines Fensters (JFrame)
- JLabel zum Anzeigen von Hello World
- Anzeigen des Fensters
- `/vol/tdpe/share/material /session11/HelloWorldSwing.java`

Das Grundgerüst

```
public class HelloWorldSwing {

    private static void createAndShowGUI() {
        ...
    }

    public static void main(String[] args) {
        //Schedule a job for the event-dispatching thread:
        //creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}
```

Fenster erzeugen und anzeigen

```
public class HelloWorldSwing {
    /**
     * Create the GUI and show it.  For thread safety,
     * this method should be invoked from the
     * event-dispatching thread.
     */
    private static void createAndShowGUI() {
        //Make sure we have nice window decorations.
        JFrame.setDefaultLookAndFeelDecorated(true);

        //Create and set up the window.
        JFrame frame = new JFrame("HelloWorldSwing");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      ...

        //Display the window.
        frame.pack();
        frame.setVisible(true);
    }
}
```

Label erzeugen und einbinden

```
public class HelloWorldSwing {
    /**
     * Create the GUI and show it.  For thread safety,
     * this method should be invoked from the
     * event-dispatching thread.
     */
    private static void createAndShowGUI() {
      ...

        //Add the ubiquitous "Hello World" label.
        JLabel label = new JLabel("Hello World");
        frame.getContentPane().add(label);

        //Display the window.
        frame.pack();
        frame.setVisible(true);
    }
}
```

Ein Clickcounter

- Ein Fenster mit einem Button
- Die Clicks auf den Button werden mitgezählt und angezeigt
- `/vol/tdpe/share/material/session11/SwingApplication.java`

Das Grundgerüst

```
public class SwingApplication {

    public static void main(String[] args) {
        //Schedule a job for the event-dispatching thread:
        //creating and showing this application's GUI.
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                createAndShowGUI();
            }
        });
    }
}
```

Fenster und Inhalte erzeugen

```
private static void createAndShowGUI() {
    ...

    //Create and set up the window.
    JFrame frame = new JFrame("SwingApplication");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    SwingApplication app = new SwingApplication();
    Component contents = app.createComponents();
    frame.getContentPane().add(contents, BorderLayout.CENTER)

    //Display the window.
    frame.pack();
    frame.setVisible(true);
}
```

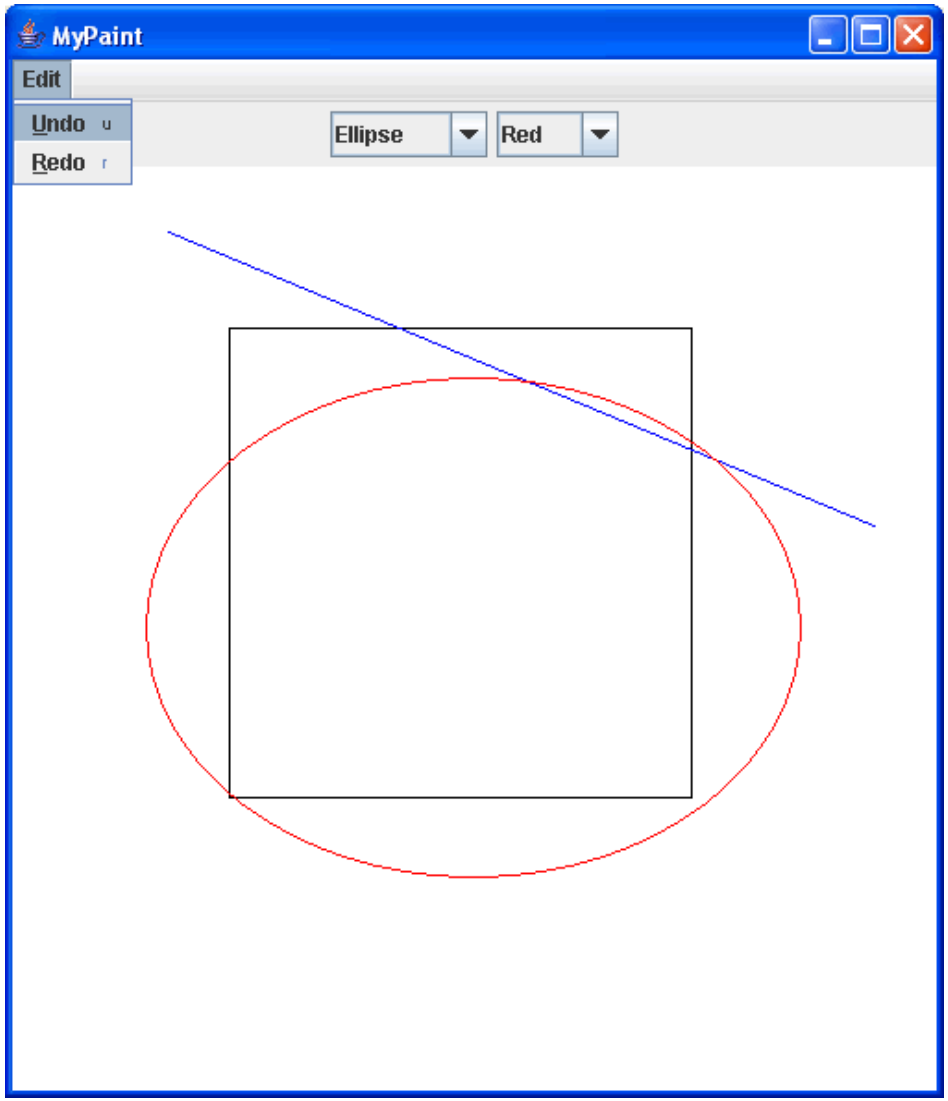
Ein einfaches GridLayout

```
private static String labelPrefix="Number of button clicks: ";
final JLabel label = new JLabel(labelPrefix + "0    ");

public Component createComponents() {
    JButton button = new JButton("I'm a Swing button!");
    button.setMnemonic(KeyEvent.VK_I);
    button.addActionListener(this);
    label.setLabelFor(button);

    JPanel pane = new JPanel(new GridLayout(0, 1));
    pane.add(button);
    pane.add(label);
    pane.setBorder(
        BorderFactory.createEmptyBorder(30, 30, 10, 30));
    return pane;
}
```

EventListener

```
public class SwingApplication implements ActionListener {

    public void actionPerformed(ActionEvent e) {
        numClicks++;
        label.setText(labelPrefix + numClicks);
    }
}
```

Aufgabe für zwei Wochen



Aufgabe für zwei Wochen

- Ein Zeichenprogramm
- Funktionen:
  - Zeichnen vor Linien, Rechtecken, Ellipsen mit der Maus
  - Je in mindest drei verschiedenen Farben
  - Der Benutzer kann Schritte rückgängig machen (UnDo)
  - Der Benutzer kann gelöschte Formen wieder einblenden (ReDo)

Aufgabe für zwei Wochen

- Abgabe: Quellcode und ausführbares Programm
- Unter: `/vol/tdpe/groupX/session11/teamY/`
- Bearbeitung zu zweit
- Zeit: zwei Wochen